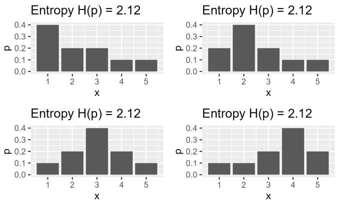
Introduction to Machine Learning

Entropy I

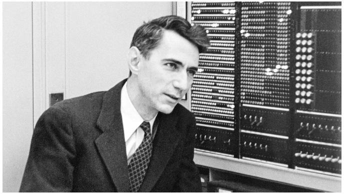


Learning goals

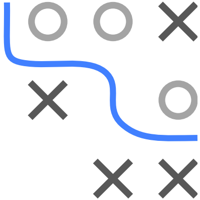
- Entropy measures expected information for discrete RVs
- Know entropy and its properties

INFORMATION THEORY

- **Information Theory** is a field of study based on probability theory.
- Foundation was laid by Claude Shannon in 1948; since then been applied in: communication theory, computer science, optimization, cryptography, machine learning and statistical inference.
- Quantify the "amount" of information gained or uncertainty reduced when a random variable is observed.
- Also about storing and transmitting information.

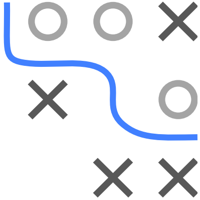


INFORMATION THEORY / 2



- We introduce the basic concepts from a probabilistic perspective, without referring too much to communication, channels or coding.
- We will show some proofs, but not for everything. We recommend *Elements of Information Theory* by Cover and Thomas as a reference for more.
- The application of information theory to the concepts of statistics and ML can sometimes be confusing, we will try to make the connection as clear as possible.
- In this unit we develop entropy as a measure of uncertainty in terms of expected information.

ENTROPY AS SURPRISAL AND UNCERTAINTY



For a discrete random variable X with domain $\mathcal{X} \ni x$ and pmf $p(x)$:

$$H(X) := H(p) = -\mathbb{E}[\log_2(p(X))] \qquad = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

$$= \mathbb{E}\left[\log_2\left(\frac{1}{p(X)}\right)\right] \qquad = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)}$$

Some technicalities first:

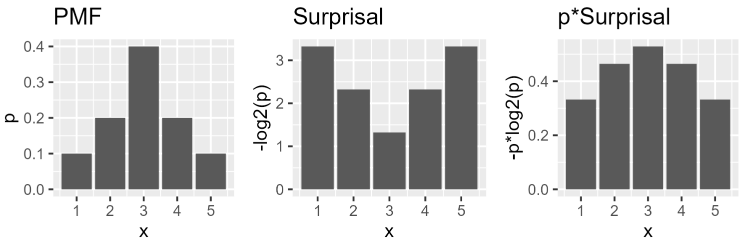
- H is actually Greek capital letter **E**ta (η) for **e**ntropy
- Base of the log simply specifies the unit we measure information in, usually bits (base 2) or 'nats' (base e)
- If $p(x) = 0$ for an x, then $p(x)\log_2 p(x)$ is taken to be zero, because $\lim_{p \to 0} p \log_2 p = 0$.

ENTROPY AS SURPRISAL AND UNCERTAINTY

$$H(X) = -\mathbb{E}[\log_2(p(X))] = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

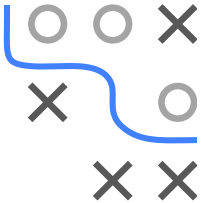
Now: What's the point?

- The negative log probabilities $-\log_2 p(x)$ are called "surprisal"
- More surprising means less likely
- PMFs surprising, so with higher H, when events more equally likely
- Entropy is simply expected surprisal



- The final entropy is $H(X) = 2.12$ (bits).

ENTROPY BASIC PROPERTIES



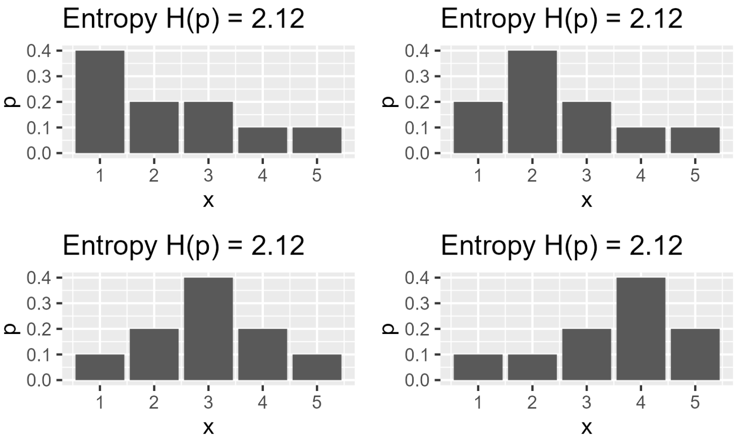
$$H(X) := H(p) = -\mathbb{E}[\log_2(p(X))] = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

1. Entropy is non-negative, so $H(X) \geq 0$
2. If one event has probability $p(x) = 1$, then $H(X) = 0$
3. Adding or removing an event with $p(x) = 0$ doesn't change it
4. $H(X)$ is continuous in probabilities $p(x)$

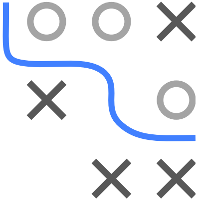
All these properties follow directly from the definition.

ENTROPY RE-ORDERING

5. Symmetry. If the values $p(x)$ in the pmf are re-ordered, entropy does not change (proof is trivial).

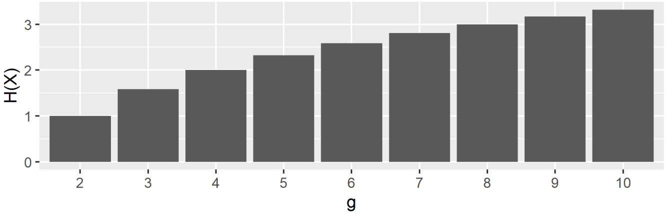


ENTROPY OF UNIFORM DISTRIBUTIONS



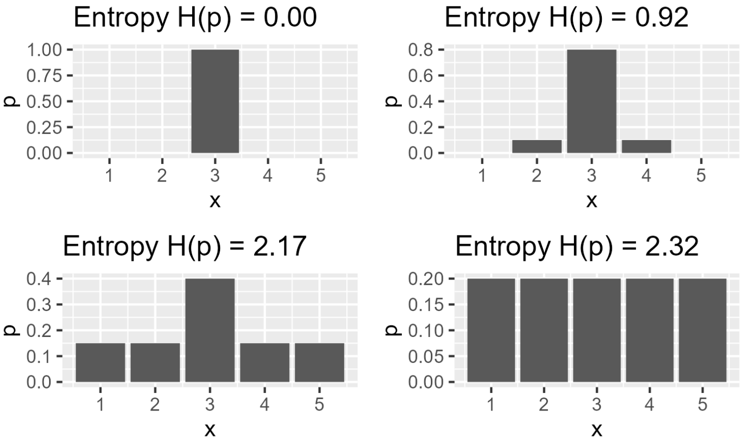
Let X be a uniform, discrete RV with g outcomes (g-sided fair die).

$$H(X) = -\sum_{i=1}^{g} \frac{1}{g} \log_2\left(\frac{1}{g}\right) = \log_2 g$$



The more sides a die has, the harder it is to predict the outcome.
Unpredictability grows *monotonically* with the number of potential outcomes, but at a decreasing rate.

ENTROPY IS MAXIMAL FOR UNIFORM



- Naive observation:
  Entropy min for 1-point and max for uniform distribution

ENTROPY IS MAXIMAL FOR UNIFORM

6. Entropy is maximal for a uniform distribution, for domain of size g: $H(X) \leq -g\frac{1}{g}\log_2(\frac{1}{g}) = log_2(g)$.

Proof: So we want to maximize w.r.t. all p_i:

$$\underset{p_1, p_2, \ldots, p_g}{\operatorname{argmax}} -\sum_{i=1}^{g} p_i \log_2 p_i$$

subject to

$$\sum_{i=1}^{g} p_i = 1$$

ENTROPY IS MAXIMAL FOR UNIFORM / 2

The Lagrangian $L(p_1, \ldots, p_g, \lambda)$ is :

$$L(p_1, \ldots, p_g, \lambda) = -\sum_{i=1}^{g} p_i \log_2(p_i) - \lambda \left(\sum_{i=1}^{g} p_i - 1 \right)$$

Solving when requiring $\nabla L = 0$,

$$\begin{aligned} \frac{\partial L(p_1, \ldots, p_g, \lambda)}{\partial p_i} = 0 = -\log_2(p_i) - \frac{1}{\log(2)} - \lambda \\ \implies p_i = \frac{2^{-\lambda}}{e} \implies p_i = \frac{1}{g}, \end{aligned}$$

last step follows from that all p_i are equal and constraint

NB: We also could have solved the constraint for p_1 and substitute $p_1 = 1 - \sum_{i=2}^{g} p_i$ in the objective to avoid constrained optimization.